

MIKROBITEN

Årgång: 6

1989

Nummer: 1

Medlemstidning för Datorklubben MIKROBITEN

GRATIS till medlemmar

Lösnummer 15 kr.

Tidningens adress:

MIKROBITEN

c/o Lennart Björk

Järnåldersringen 422. 136 65 HANDEN

INNEHÅLL:

RAPPORT FRÅN ÅRSMÖTET

FRÅGESPALT

ÖVERFÖRING MICROBEE - PC

PROGRAMMERING

ORDBEHANDLING MED WORDSTAR

MICROBEE I SKOLAN

samt listningar för Pascal och Assembler

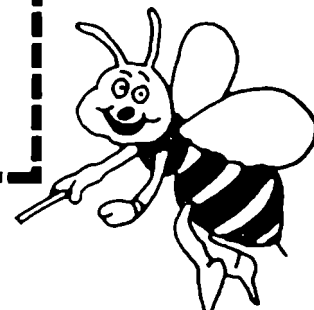
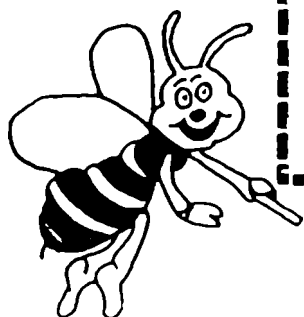
KLUBBMÖTEN

30 MARS, 27 APRIL & 18 MAJ

Lokal: TRYGG HANSA Flemminggatan 18

Tid: 18.30

Klubbens postgiro: 94 53 52-3



I N N E H A L L

| | | | |
|----------------------------------|----|-------|---|
| Ledare | 2 | | |
| Programkopiering | 3 | | |
| Arsmötet | 4 | | |
| Frågor & svar | 10 | | |
| IBM kontra Microbee | 12 | | |
| Ordbehandling med WordStar | 13 | | |
| * RST.PAS | 14 | (W01) | D |
| * RST.ZSM | 14 | (W02) | D |
| Kort om nya program | 15 | | |
| * BILREGN.MWB | 15 | (W03) | D |
| * STOREBOK.MWB | 15 | (W04) | |
| * CPM-DOS.COM | 15 | (W05) | D |
| Microbee i skolan | 15 | | |
| överföring Microbee - PC | 23 | | |
| Stridsflygaren | 24 | | |
| Programmering | 24 | | |
| Nästa nummer | 38 | | |

Asterisk (*) markerar program.

K L U B B I N F O R M A T I O N

K L U B B E N S T E L E F O N N U M M E R

| | | |
|---|-------------------|----------------|
| Medlems ärenden | : Urban Nielsen | 08 - 96 06 41 |
| Tidningen och övriga ärenden | : Lennart Björk | 08 - 777 47 62 |
| Hårdvara och Assembler | : Olle Ljungkvist | 08 - 732 79 72 |
| Hjulkvara och Kommunikation | : Carl Sundbon | 08 - 717 79 07 |
| KONsystemet Chaos (300/1200/2400 & 1200/75 dygnet runt) | : Carl Sundbon | 08 - 717 91 80 |

K L U B B E N S A D R E S S E R

| | |
|-----------------------|--|
| Tidningen | : Mikrobiten, c/o Lennart Björk, Järnåldersringen 422, 136 65 Handen |
| Klubb Program | : Per-Didrik Orling, St. Björnens Gata 132, 136 44 Haninge |
| Public Domain Program | : Lennart Björk, Järnåldersringen 422, 136 65 Handen |
| Kassett Biblioteket | : Carl Sundbon, Vasavägen 5, 133 33 Saltsjöbaden |

K L U B B A V G I F T E R

| | |
|-----------------------|----------------------|
| Klubb Program | : 10 Kr per program. |
| Public Domain Program | : 30 kr per diskett. |
| Kassett Biblioteket | : 10 kr per kassett. |

Beställningar sker genom att sätta in pengar på klubbens postgiro 94 53 52-3 och skicka ett brev, med en beställning av de program, disketter eller kassetter som önskas, till respektive administratörs adress (se ovan). Ange, om möjligt, program- eller diskettkod. Lånetiden för kassetter är en vecka. Glöm inte att bifoga ett frankerat självadresserat kuvert vid beställning av klubbprogram!!

MIKROBITEN LEVER, MIKROBITEN FORTSÄTTER!

Än så lever Microbee och dess användarklubb i Sverige, Mikrobiten. Detta trots att tillverkningen av microbeedatorerna upphörde i Australien för ett år sedan. Så bra och så omtyckt är den så att den fortfarande är efterfrågad och hamnar hos nya användare. Kommersen på begagnade maskiner är stor. Det gäller alla modeller. Och så vitt jag vet blir alla sålda som är till salu. Man får ställa sig på väntelista hos Network AB i Kungsbacka om man önskar en Premium eller 256:a. Network fortsätter alltså att ha begagnatmarknad på microbeeorrådet. Många har utnyttjat det och har sedan blivit medlemmar i vår klubb, om de inte har varit det tidigare. Så även om vi har förlorat en och annan medlem som har fallit offer för "det hemska IBM PC/MS-DOS" så tillkommer nya i stället. Intresset för klubben är stort. Det tackar jag för, samtidigt som jag hälsar de nytillkomna hjärtligt välkomna i klubbgemenskapen.

Jag vill här också tacka för alla bidrag till vårt blad, som har kommit på sistone. Jag ber om ursäkt att jag inte har hunnit tacka alla personligen, men hoppas att Ni här skall känna Er ihågkomna.

Lennart Björk

ARSAVGIFTEN 1989

Med detta tidningsnummer medföljer ett inbetalningskort för 1989 års medlemsavgift. Det skall Du använda för att betala årets avgift innan Du utnyttjar fler medlemsförmåner såsom programbeställningar och kassettlån. Du bör också betala avgiften så snart som möjligt för att inte riskera att gå miste om nästa nummer av tidningen, utsändningen avbryts nämligen för den, som inte betalar. Det bör heller inte vara svårt att göra oss till viljes, då medlemsavgiften är oförändrade låga 75 kronor. Samma avgift för fjärde året i rad. Du, som redan har betalt för 1989 behöver inte bry Dig om ovanstående.

Till Er som är lärare, och som önskar att Er skola skall betala avgiften, vill vi säga att det bästa är att Ni själva lämnar inbetalningskortet till ekonomiavdelningen och ber den att ange Ert namn som referens. Vi har nämligen svårigheter att pussla ihop person, som skall ha tidningen, med kommun som har betalt avgiften! Ibland har vi inget namn alls. Det säkraste sättet för Dig som är lärare att få tidningen torde vara att Du själv lägger ut de 75 kronorna och tar postens kvitto med till skolans/kommunens kontantkassa och ber att få ersättning för det utlagda. Detta sagt som ett förslag.

ordf. o. kassör

P.S. Se till att Ditt namn står på inbetalningskortet!, så att det otrevliga inte inträffar att Du har betalt utan att vi vet.

På vårt klubbmöte på Trygg Hansa den 30:e mars kan det hända att vi får en demonstration av IBM-kompatiblern Euro-PC, ett av NETWORKS i Kungsbacka svar på MS-DOS-fronten. Det blir i så fall folk från just Network som står för demon. Vi skall också göra ett försök att sätta ihop en redaktion för Mikrobiten, så kom alla ni, som har synpunkter på Mikrobitens framställning, eller möjligheter att hjälpa till på ett eller annat sätt!

Ä L D R E N U M M E R A V M I K R O B I T E N

Det går att beställa äldre nummer av Mikrobiten. Priset är 15 kr per ex. eller 75 kr per årgång. Betalning görs på vanligt sätt till klubbens postgiro varefter beställning sänds till klubbens officiella adress (Lennart Björk, Järnåldersringen 422, 136 65 Haninge).

P R O G R A M K O P I E R I N G

Nu är det dags att betala ut pengar till klubbens programmakare. Av det tio kronor som beställaren av ett klubbprogram betalar går fem kronor till den som har gjort programmet. Under 1988 har det beställts totalt 169 klubbprogram så det är alltså 845 kronor som skall betalas ut.

Eftersom det kostar klubben fem kronor att sända pengar till en person vars Postgiro eller Personkonto nummer vi inte känner till så vore det bra om ni som har ett sådant konto (och har sänt in klubbprogram) kunde höra er av till klubben och meddela vilket kontonummer ni har. Detta görs enklast på ett av Postens adressändringskort.

Vi har också beslutat att vänta med utbetalningarna till de medlemmar vars program har inbringat mindre än 25 kronor (5 kopior) i år. Dessa pengar kommer istället ackumulera sig och betalas ut när de nått den magiska summan.

Vi vill också passa på och tacka de medlemmar som donerar sina kopieringsavgifter till klubben och på det sättet bidrar till klubbens goda ekonomi.

VERKSAMHETSBERÄTTELSE FÖR 1988

Under året har medlemsantalet ökat från 334 till 378 betalande medlemmar. En rekryteringskampanj riktad framförallt till landets lärare har varit framgångsrik. Antalet klubbmöten har varit 7. Styrelsen har dessutom sammanträtt 5 gånger och hårdvarugruppen har mötts 4 gånger. Tidskriften Mikrobiten har utkommit med 5 nummer. Innehållet har dels varit program, gjorda av medlemmarna själva, dels artiklar och nyheter. En uppskattad kurs i assembler har gått som serie i ett antal nummer. Under året har försök gjorts med modern tidningsteknik, s.k. desktop publishing. Försöket har dock lagts ned p.g.a. bristande kundkaper och resurser.

Vid årsmötet den 21 januari 1988 utsågs följande styrelse och funktionärer:

| | |
|----------------|-----------------|
| Ordförande | Lennart Björk |
| v. ordförande | Olle Ljungquist |
| Kassör | Urban Nielsen |
| Sekreterare | Teodor Canback |
| v. sekreterare | Daniel Grönjard |

Till revisorer för 1988 valdes Harry Dackeby och Per-Didrik Örling. Till valberedning för tiden fram till nästa års möte valdes Carl Sundbom och Peter Bäckström.

Klubben har under året förlorat en stor kraft. Nämligen eldsjelen Nils Bildsten, som praktiskt taget vigdesista delen av sitt liv åt microbeedatorn och dess största svenska klubb. Det var Nils, som drog upp Mikrobiten från 80 medlemmar 1985 till 600 på två år. Han var en stor medlemsvarvare och man kan nog säga, att det var hans förtjänst, att Mikrobiten blev den enda kvarvarande microbeeklubben i Sverige av de tre, som har funnits. De andra klubbarnas medlemmar har i stor utsträckning anslutit sig till vår förening Mikrobiten. Nils skötte dessutom allt, som hade med klubben att göra, medlemsservice, programdistribution, experthjälp m.m. Dessutom använde han sina gedigna journalistiska kunskaper och producerade och redigerade vår tidskrift Mikrobiten helt själv. Nils fann livsgnoge i klubbarbetet och de många medlemskontakterna höll honom vid gott humör trots hans cancer. Det inspiherande skrivandet för vår tidning fick honom att för tillfället glömma värken och sjukdomen. Detta och det faktum att Nils de sista åren dels levde ensam, dels var sjukskriven, först på halvtid senare på heltid, gjorde att han hade möjlighet att ägna klubben så mycket tid. När Nils togs in på sjukhus alldeles i början på somaren, hade han trotsat sjukdomen i flera år. Några dagar senare hade Nils liv flytt. Släkt och vänner hade förlorat en mycket omtrodd människa. Dessförinnan, några månader tidigare, uppvaktade klubben Nils Bildsten i samband med hans 60-års dag. Det arbete som Nils utförde för klubben, har fördelats på ett antal klubbmedlemmar. Förändringarna har medfört en i vissa fall senare medlemservice, trots att styrelsen försökt mildra konsekvenserna av det inträffade.

Det finnes ingen, som i kapacitet kan mata sig med Nils

när det gäller insatser för klubben. När han vila i frid.

Lennart Björk Olle Ljungquist Urban Nielsen Daniel Grönjard
Teodor Canback

INTAKTER

| | | | |
|--------------------------|------|------------|----------|
| 300 MEDLEMSAVGIFTER | 0.00 | 24075.00 | 24075.00 |
| SUMMA KLASS 30 | 0.00 | 24075.00* | 24075.00 |
| 310 INTAKT PROGRAMKOPIOR | 0.00 | 3370.20 | 3370.20 |
| SUMMA KLASS 31 | 0.00 | 3370.20* | 3370.20 |
| 320 INTAKT KASSETTLAN | 0.00 | 511.40 | 511.40 |
| SUMMA KLASS 32 | 0.00 | 511.40* | 511.40 |
| 330 INTAKT PD-PROGRAM | 0.00 | 360.00 | 360.00 |
| SUMMA KLASS 33 | 0.00 | 360.00* | 360.00 |
| 340 RANTEINTAKTER | 0.00 | 1952.99 | 1952.99 |
| SUMMA KLASS 34 | 0.00 | 1952.99* | 1952.99 |
| SUMMA INTAKTER | 0.00 | 30269.59** | 30269.59 |

KOSTNADER

| | | | |
|--------------------------|------|-------------|-----------|
| 420 UTGIFTER HARDVARUGRU | 0.00 | -3033.00 | -3033.00 |
| SUMMA KLASS 4 | 0.00 | -3033.00* | -3033.00 |
| 610 TIDN MIKROBITEN | 0.00 | -13222.90 | -13222.90 |
| 620 PORTON | 0.00 | -2213.00 | -2213.00 |
| 630 ÖVRIG ADMINISTRATION | 0.00 | -4405.00 | -4405.00 |
| SUMMA KLASS 6 | 0.00 | -19840.90* | -19840.90 |
| 710 AVSKRIVNING INVENTAR | 0.00 | -1770.00 | -1770.00 |
| SUMMA KLASS 7 | 0.00 | -1770.00* | -1770.00 |
| SUMMA KOSTNADER | 0.00 | -24643.90** | -24643.90 |
| RESULTAT | 0.00 | 5625.69** | 5625.69 |

HELA PERIODEN

Högsta ver.nr: 382

VISAR SALDOBELOPP

KONTO KONTOBENÄMNING

ING. SALDO

DEBET

KREDIT

UTG. SALDO

| | | | | | |
|-----|----------------------|-----------|----------|----------|-----------|
| 100 | POSTGIRO | 25009.80 | 32569.09 | 36313.04 | 21265.85 |
| 110 | KONTANTKASSA | 0.00 | 75.00 | 75.00 | 0.00 |
| 192 | INVENTARIER | 2360.00 | 0.00 | 1770.00 | 590.00 |
| 200 | SKULDER | -16089.64 | 16089.64 | 4800.00 | -4800.00 |
| 210 | MEDLEMSAVGIFTER 1989 | 0.00 | 0.00 | 150.00 | -150.00 |
| 240 | FÖREG ARS VINST | -1394.26 | 0.00 | 0.00 | -1394.26 |
| 250 | BALANSERAD VINST | -9885.90 | 0.00 | 0.00 | -9885.90 |
| 300 | MEDLEMSAVGIFTER | 0.00 | 4800.00 | 28875.00 | -24075.00 |
| 310 | INTÄKT PROGRAMKOPIOR | 0.00 | 0.00 | 3370.20 | -3370.20 |
| 320 | INTÄKT KASSETTLAN | 0.00 | 0.00 | 511.40 | -511.40 |
| 330 | INTÄKT PD-PROGRAM | 0.00 | 0.00 | 360.00 | -360.00 |
| 340 | RÄNTEINTAKTER | 0.00 | 0.00 | 1952.99 | -1952.99 |
| 420 | UTGIFTER HARDVARUGRU | 0.00 | 3033.00 | 0.00 | 3033.00 |
| 610 | TIDN MIKROBITEN | 0.00 | 13290.90 | 68.00 | 13222.90 |
| 620 | PORTON | 0.00 | 2298.10 | 85.10 | 2213.00 |
| 630 | ÖVRIG ADMINISTRATION | 0.00 | 7001.40 | 2596.40 | 4405.00 |
| 710 | AVSKRIVNING INVENTAR | 0.00 | 1770.00 | 0.00 | 1770.00 |
| | ***SALDOTOTALER*** | 0.00 | 80927.13 | 80927.13 | 0.00 |

BALANSRAPPORT

MIKROBITEN

DATUM: 890214 SID 1

HELA PERIODEN

Högsta ver.nr: 382

| | | ING. SALDO | SALDO | UTG. BALANS | |
|--------------|-----------------------------|------------|-----------|-------------|-----------|
| TILLGANGAR | | | | | |
| 100 | POSTGIRO | 25009.80 | -3743.95 | 21265.85 | |
| 192 | INVENTARIER | 2360.00 | -1770.00 | 590.00 | |
| | SUMMA TILLGANGAR | *** | 27369.80 | -5513.95 | 21855.85 |
| SKULDER | | | | | |
| 200 | SKULDER | -16089.64 | 11289.64 | -4800.00 | |
| 210 | MEDLEMSAVGIFTER 1989 | 0.00 | -150.00 | -150.00 | |
| 240 | FÖREG ARS VINST | -1394.26 | 0.00 | -1394.26 | |
| 250 | BALANSERAD VINST | -9885.90 | 0.00 | -9885.90 | |
| | SUMMA SKULDER | *** | -27369.80 | 11139.64 | -16230.16 |
| EG - KAPITAL | | | | | |
| | BERÄKNAT EGET KAPITAL DIFF | *** | 0.00 | -5625.69 | -5625.69 |
| | SUMMA SKULDER OCH EGET KAP. | *** | -27369.80 | 5513.95 | -21855.85 |

**MIKROBITEN DATORKLUBB
ÅRSMÖTET 16 FEBR 1989**

REVISIONSBERÄTTELSE

för

Verksamhetsåret 1988

Undertecknade, som valts av årsmötet 1988 att granska klubbens räkenskaper för år 1988, avlämnar följande revisionsberättelse.

Nils Bildstens bortgång i juni 1988 innebar på flera sätt en stor förlust för klubben. Bl annat brast kontinuiteten i den ekonomiska uppföljningen. Styrelsen har här lagt ned ett förtjänstfullt arbete för att rekonstruera redovisningen för speciellt mars 1988.

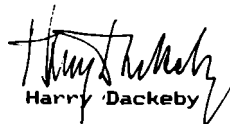
Då det gått att framställa ett fullt tillfredsställande underlag för granskningsarbetet vill vi tillstyrka

- att resultaträkningen och balansräkningen fastställs
- att vinsten disponeras i enlighet med styrelsens förslag i verksamhetsberättelsen samt
- att styrelsens ledamöter beviljas ansvarsfrihet för år 1988.

Stockholm den 14 febr 1989


Per-Didrik Örling

Revisorer


Harry Dackeby

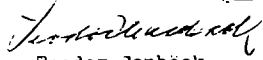
Protokoll fört vid årsmöte den 16 febr. 1939.
Närvarande: 12 medlemmar.

- § 1 Årsmötet öppnades av klubbens ordförande Lennart Björk.
- § 2 På fråga om stadgeenlig kallelse skett, blev svaret: ja.
- § 3 Den föreslagna dagsordningen godkändes utan tillägg.
- § 4 Till mötesordförande valdes Lennart Björk.
- § 5 Till mötessekreterare valdes Teodor Sanbäck.
- § 6 Till justeringsman valdes Peter Beckström och Per-Didrik Orling.
- § 7 Beslöts att dagens närvarolista skulle användas som röstlängd.
- § 8 Styrelsens förvaltnings- och verksamhetsberättelse genomtogs.
- § 9 Den ekonomiska berättelsen föredrogs av Urban Nielsen och kompletterades av Per-Didrik Orling.
- § 10 Revisionsberättelsen föredrogs av Per-Didrik Orling.
- § 11 Styrelsen beviljades ansvarsfrihet för verksamheten under 1938.
- § 12 Styrelsen föreslog oförändrad medlemsavgift för 1939 vilket också blev årsmötets beslut, d.v.s. 75:- per år.
- § 13 Ingen direkt avsägelse från den sittande styrelsen förelåg men Olle Ljungquist hade antytt att han ville minska sitt engagemang på grund av tidsbrist.
- § 14 Efter val fick den nya styrelsen följande sammansättning:
Ordförande : Lennart Björk
v. ordf.: Daniel Grönjord
Kassör: Urban Nielsen
sekreterare: Teodor Sanbäck
v. sek.: Carl Sundbo
- § 15 Till revisorer valdes: Harry Dackeby och Per-Didrik Orling.
- § 16 Till valberedning valdes: Peter Beckström och Daniel Sundkvist.
- § 17 Övriga frågor:
 - a) Diskuterades en ändring av stadgarna. Ingen ändring besluts.
 - b) Peter Beckström lovade att undersöka om det fanns möjlighet att till klubben anskaffa en skrivare till billigt pris.
 - c) Lennart Björk efterlyste frivillig arbetskraft till tidskriften för bl.a. redigerings- och skrivarecte av översikts-karaktär.
 - d) Framfördes förslag att i tidskriften införa en "frågöklad" och en "undomssida".

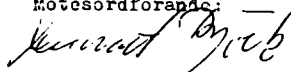
§ 18 Per-Didrik Orling framförde Ersmötets tack till styrelsen för arbetet under 1988.

§ 19 Avslutades Ersmötet.

Vid protokollet:


Teodor Janbäck

Mötesordförande:


Lennart Björk

Justeras:


Peter Beckström


Per-Didrik Orling

FRÅGOR & SVAR

Eftersom det kommer in en hel del brev med frågor och tips av allehanda slag till redaktionen så har vi från och med detta nummer ett speciellt utrymme reserverat för detta. De frågor vi inte har kunnat svara på litar vi på att klubbens samlade kunskap skall kunna reda ut. Lösningar och tips sänds med fördel också till redaktionen så att vi kan ta med dem i det påföljande numret av tidningen.

Ingrid Maier i Uppsala vill ha hjälp med Graphic WordBee i kassett versionen.

"Jag undrar om det är någon som kan hjälpa mig med några alfabet i kassettform, främst grekiska, tjeckiska och fonetiska tecken. I gengäld kan jag stå till tjänst med det ryska och det polska alfabetet.

Sedan undrar jag om det finns någon som kan hjälpa mig med ett tips, nämligen hur man kan modifiera ett existerande alfabet. Det behövs t.ex. bara några smärre ändringar i det ryska alfabetet plus två-tre nya tecken för att man skall kunna skriva texter på t.ex. serbokratiska, vitryska, ukrainska eller azerbajdanska. Men jag klarar inte av att ändra minnesinnehållet i önskat riktning. Den font editor som jag har hjälper inte heller så mycket eftersom jag inte kan lägga in det ursprungliga ryska alfabetet, utan där finns bara det vanliga svenska alfabetet som utgångspunkt. Och då blir det ett enormt tidskrävande arbete att skapa alla tecknen på nytt!

Jag är mycket tacksam för alla tips och står naturligtvis också själv till förfogande med mina egna erfarenheter."

Ingrid Maier
Döbelnsgatan 24 C
752 37 Uppsala
018-14 52 34 (kvällstid)

Från Rolf T. Andersen i Norge har vi fått denna förfrågning. Rolf nämner att man kan trycka på ESC för att aktivera S3DUMP men oftast används Ctrl-P.

"Jeg har tidligere brukt skjermdumpen S3DUMP i en del av mine program. Denne rutinen var fin med at den ikke var innebygd i hovedprogrammet. Det var bare nødvendig å trykke ESC og velge skrivertype, så var det hele i gang. Feilen med denne rutinen er at den ikke går særlig bra sammen med PREMIUMENS grafikk. Jeg har prøvd å få programmet PREMDUMP.MWB til å fungere på samme måten, men uten å lykkes. Jeg regner med at dette er fullt mulig. Det er nok mine kunnskaper på dette feltet som svikter. Derfor: Hvordan får en PREMDUMP-rutinen til å starte bare ved et trykk på ESC hvor som helst i et program?"

Med hilsen
Rolf T. Andersen
Kjerkelia 6
8650 Mosjøen, Norge

Reidar Tolläng i Göteborg har några frågor om hårdvara och ett tips angående Sei-15 skrivare.

"I samband med köp av ny skrivare upptäckte jag att det inte finns någon samlad beskrivning av vad Microbee kräver av utrustning som skall anslutas, detaljer om koder, IBM och Epson, databitar, kablar för seriell och parallell matning, olika kontakter, med eller utan inbyggd elektronik, osv. Finns det möjligen någon tekniskt bevandrad medlem som kan göra en enkel och överskådlig tablå för de olika Microbee modellerna."

Reidar undrar också hur man kan koppla in ett bättre tangentbord till Microbee. Detta är en klassiskt problem som jag inte tror att det finns någon bra lösning på, men vi tar gärna emot tips om hur några förbättringar skulle kunna ske.

Ägare av Sei-15 skrivare kan i framtiden få det svårt med färgband men här kommer en lösning på det problemet.

"Häromsistens fick jag veta av Network att fabriken som tillverkade Sei-15 skrivarna har gått i konkurs och att Networks lager av färgband är slut och inte kan förnyas. För Sei-15 ägare som av någon anledning har sparat tomma färgbandskassetter är det inget större problem - det är relativt enkelt att med små modifieringar ladda om dem med färgband för t.ex. CPA-80."

Reidar Tolläng
Perukmakargatan 13
415 06 Göteborg

Från Michael Bafford i Kristianstad, som inte är medlem i klubben (ännu), har vi fått några frågor som flera kanske har funderat på. Michael använder sin Microbee 64 till ordbehandling men har på senare tid fått upp ögonen för andra användningsområden och att Microbee inte är den allra modernaste datorn på marknaden. Han skriver bla. så här:

"CP/M (Microbee) verkar vara akterseglat - kanske snarare förbigången. Och det är synd, för denna dator klarar av så mycket mer än jag behöver att jag tycker det är skam att operativa systemet är förlegat. För det är det väl?

Det finns så lite spel för pjäsen. Att ta datorn till ren förströelse är nästan en förutsättning för att jag inte skall börja betrakta den som ett verktyg - ett arbetsredskap. Och särskilt nu då det finns så många intressanta spel; interaktiva deckare t.ex. och flygsimulatorer...

Jag har också hört att det finns ett program som gör Microbee kompatibel med en PC; åt något håll."

CP/M är utan tvekan på väg att bli ett avslutat kapitel, men det säljs faktiskt en del CP/M datorer fortfarande. Och precis som man kan köra en gammal bil så kan man förstås köra en gammal dator; det är inte alla som behöver turbo och låsningsfria bromsar.

Det har gjorts en hel del spel till Microbee. Problemet är att de kan vara svåra att få tag i. Klubbens kassett bibliotek inne-

håller en hel del bra spel och det finns också några bra klubb och PD program som t.ex. Benjakt, Masken och Brickball. Network kan också erbjuda samlingar av spel på speciella speldisketter. Infocom har gjort några helt superba äventyrsspel för CP/M (Microbee) som t.ex. Hitchhiker och Zork. Dessa såldes av Network förut men hur det är med det nu vet jag inte. I varje fall finns äventyrsspelet Colossal Cave i klubbens Public Domain bibliotek och det är i samma klass.

Att göra Microbee PC kompatibel är omöjligt men man kan flytta (text) material mellan dessa två maskiner med ett program som heter PCBEE. PCBEE går att beställa ifrån klubben. (Se även artikeln "Överföring Microbee - PC" i detta nummer)

Slutligen vill redaktionen ställa en fråga som vi hoppas kunna få svar på till nästa nummer. Som en del kanske har lagt märke till så finns det inga checksummor till Basic programmen längre. Detta beror på att tekniken att kunna lista ett program inklusive checksummorna till en fil inte är känd här på redaktionen.

I B M K O N T R A M I C R O B E E

SKATTA ER LYCKLIGA, NI MICROBEE (CP/M) ÄGARE!

Jag vet inte om jag ska skratta eller gråta. Efter att ha kört microbee ända sedan den kom till Sverige, tillfogat program efterhand som de kommit och varit mer än nöjd med resultatet med vad det kunnat prestera - har jag nu av en ren tillfällighet skaffat en IBM, inte en kompatibel utan en ren IBM. Vad tycker jag då om denna föregångare till så många efterapningar? Ja det kanske rubriken skvallrar om.

För att ta det från början upptäckte jag att då man formatterat en skiva och lagt in systemet på den så fanns det endast knappt 300 kb kvar. Alltså om man jämför med Microbee har man redan här tappat mer än 90 kb. Om man i CP/M (microbee) kör under Shell har man på disketten 386 kb - shell:s 6 kb = 380 kb fritt, då hela systemet + Shell finns inlagt.

Andra erfarenheten var kalkylprogrammet Multiplan. På microbee upptar det 110 kb och på IBM PC-DOS upptar det totalt 239 kb! detta ger, att i microbee har du 129+90 kb mer kvar på disken och att den kräver också nära 130 kb mindre i minne.

På en microbee disk kan du lägga in Multiplan 3 gånger på samma disk, även om det är heltokigt, och ändå ha ett utrymme kvar på 56 kb, på en IBM eller kompatibel kan du endast lägga in det en gång på en och samma disk, och då har du ca 50 kb kvar på denna. Helt klart är väl, tycker jag, att IBM systemet kräver en hård-disk om man ska gå in för systemet på allvar.

Till sist tar det en evinnerlig tid att starta upp datorn, man kan nästan ta en fikapaus då man slagit på strömbrytaren tills det finns någon prompt på skärmen.

Ja det finns nog många som tycker olika saker på olika sätt då det gäller datorer, detta var lite funderingar som jag fick efter en ny bekantskap - Och den har då inte fått mig att kasta elaka blickar på min microbee - tvärtom så tycker jag nu mer om min 128:a - Gammal kärlek rostar aldrig.

Datahälsningar från en lycklig microbeeanvändare

Jan Pettersson

ORDBEHANDLING MED WORDSTAR

I förra numret så kom Frank Lindgren med ett tips om hur man kan få fram gamla filnamn i WordStar med hjälp av att trycka på CTRL-D ett flertal gånger. I själva verket så är det lättare än så, tryck bara CTRL-R så kommer hela filnamnet upp direkt. Detta är idealiskt när man kör andra program från WordStar och sedan vill återgå och arbeta på samma text igen.

ibland märker man att vissa program inte går att använda ifrån WordStar på grund av att det inte finns tillräckligt med minne ledigt. Detta fenomen kan inträffa fastän programmet är väldigt litet och det beror med stor sannolikhet på att programmet är skrivet i Turbo Pascal. Vad som händer är att det blir en konflikt mellan Turbo Pascal och WordStar som båda vill använda en liten minnesdel mycket högt upp i datorns minne. Detta problem kan lösas på två olika sätt.

För nybörjaren är det kanske lättast att kompilera Turbo Pascal programmen ifrån WordStar. Dvs. först starta WordStar och trycka 'R' ifrån huvudmenyn. Skriv sedan TURBO (förutsatt att man har Turbo Pascal på samma diskett) och sedan är det bara att fortsätta som vanligt. Skillnaden är att Turbo Pascal'n har anpassat sig till det minnesutrymme som WordStar lämnat kvar.

En annan metod, som ger exakt samma effekt, är att i Turbo Pascals 'Options' meny ('kompileringsval' i svenska Turbo Pascal tror jag) ändra den övre adressen för kompileringen till CF42. Denna adress gäller för Turbo Pascal 3.01A samt WordStar 3.30 men den bör fungera för alla de andra versionerna också eftersom de kräver mindre minnesutrymme. I annat fall så kan man kompilera ifrån WordStar som beskrivs ovan.

När man har använt vissa program ifrån WordStar som innehåller grafik, t.ex. TRANSFER, så finner man att nästan alla text som normalt är i invers nu har bytts ut mot små delar av den grafik som fanns i programmet man använde. Nedan är ett kort Turbo Pascal program listat som man kan använda för att återställa texten till normalt utseende. Eftersom den största delen av programmet egentligen är ren maskinkod så kan det med fördel skrivas direkt i assembler. Programmet kan användas till Turbo Pascals editor för samma ändamål.

Daniel Grönjörd

PROGRAM Rst;

BEGIN

```
    WriteLn('RST v1.0 (c) 1989 Daniel Grönjörd');
    INLINE (H3E/01/ (* LD A,1 *)
           HD3/H0B/ (* OUT (11),A *)
           H11/HF800/ (* LD DE,F800H *)
           H21/HF000/ (* LD HL,F000H *)
           H01/H0800/ (* LD BC,0800H *)
           H7E/ (* LOOP: LD A,(HL) *)
           H2F/ (* CPL *)
           H12/ (* LD (DE),A *)
           H13/ (* INC DE *)
           H23/ (* INC HL *)
           H0B/ (* DEC BC *)
           H79/ (* LD A,C *)
           HB0/ (* OR B *)
           H20/HF6/ (* JR NZ,LOOP *)
           H3E/H00/ (* LD A,0 *)
           HD3/H0B) (* OUT (11),A *)
END (* Rst *).
```

Här är assembler varianten.

```
    ORG    0100H
;
;Skriv ut meddelandet
;
    LD     C,9
    LD     DE,MSG
    CALL  5
;
;Återställ tecknen
;
    LD     A,1
    OUT   (11),A
    LD     DE,0F800H
    LD     HL,0F000H
    LD     BC,0800H
LOOP:   LD     A,(HL)
    CPL
    LD     (DE),A
    INC   DE
    INC   HL
    DEC   BC
    LD     A,C
    OR    B
    JR    NZ,LOOP
    LD     A,0
    OUT   (11),A
    RET
;
MSG:   DEFB   'RST v1.1 (c) 1989 Daniel Grönjörd'
;
    END
```

De program som det inte fanns plats för i tidningen ges under den här rubriken en kort presentation. De kan förstås beställas precis som alla andra klubbprogram.

Från Rolf T. Andersen har vi fått två nya program. Det första är en ny version av programmet BILREGENSKAP (också av Rolf). Så här beskriver Rolf ändringarna: Det är nå mulig å få regnskapsöversikt for de enkelte måneder, eller eventuelt år. Du kan nå søke fra oppgitt dato uten å vite posteringsnummer. Dessuten er det lagt inn kontrollrutiner for innskriving av dato. Det er litt viktig at dette gjøres riktig, da datoen styrer utskriftsbildet på skriveren, og for å få søkerutinene til å fungere riktig.

Det andra programmet heter STOREBOK och det skriver ut bandroller med hjälp av mycket stora bokstäver. Detta kan då ske över flera ark eftersom de skrivs ut 'liggande'.

Ett program som konverterar texter mellan vanlig ASCII (dvs. Microbee) och IBM teckenuppsättning är CPM-DOS av Daniel Grönjörd. PCBEE klarar av denna konvertering i normala fall men överför man texterna på ett annat sätt (t.ex. via modem) så måste denna konvertering göras med ett separat program. Programmet kräver Turbo Pascal 3.01A och finns därför bara i kompilerad form.

M I C R O B E E I S K O L A N

Dagens skolspalt innehåller två artiklar: dels fortsättningen på mät- och styr från förra numret och dels en artikel om datorn i språkundervisningen.

Just nu skrivs det mycket om copyright på datorprogram och om piratkopiering. Många lärare gör egna program och vill givetvis inte skänka bort dem till andra skolor. Det är därför viktigt att vi lärare inte använder fria klubbprogram i undervisningen, utan köper dem ordentligt, direkt från upphovsmannen. Oftast kostar programmen bara någon hundralapp. Att du provar programmen hemma är OK, men använder du dem i arbetet, se till att din arbetsgivare gör rätt för sig!

I förra numret fanns ett bidrag av Lars-Einar Abrahamsson om datorn på mellanstadiet. Nu har jag lyckats ladda in det ena av de program han skrev om, SKOLAN.MWB, ett äventyrsspel av god klass. Det går att köpa (i kassett) tillsammans med programmet TEXT1 (vilket jag pga bandfel ej lyckats läsa) direkt från:

Lars-Einar Abrahamsson
Kövra 3194
840 44 OVIKEN
Tel: 0643-401 31

Ragnar Sunesson från Tranås, tillhör "Jönköpingslänsmaffian" av datalärare. I slutet av förra året, bad jag honom skriva om sina erfarenheter av datorn i språkundervisningen. Dessutom bad jag att få se på hans datorprogram.

Ragnar skriver många av de program han använder, och dessutom mycket bra program. Jag har roat mig med att försöka klara av tyska böjningar och att undvika att bli hängd på grund av dålig ordkunskap i engelska. Fast inte ens Ragnars spel lyckades få mig att tycka att tyska kan vara roligt. Med engelska gick det förstås bättre, det var rent av riktigt roligt! Tur att datorlitteratur är på engelska och inte tyska!

Efter några timmars arbete med såväl undervisningsspel som rena träningsprogram, kan jag inte annat än rekommendera programmen. Dom är omväxlande och trevliga. En viss långsamhet finns ibland, men .MWB-basiceen är klumpig att jobba med när programmen blir stora. Överlag märks det att särskilt spelen är gjorda av en kreativ person. Du som vill pröva på några nya språkprogram kan kontakta Ragnar direkt:

Ragnar Sunesson
Februarigatan 3
573 38 TRANÅS
Tel: 0140-111 52

DATORSTÖD I SPRÅKUNDERVISNINGEN

Först en kort presentation av författaren: Lärare i språk som egentligen alltid varit mer intresserad av naturvetenskap. Real-linjen på gymnasiet, matte ett år vid universitet men sadlade om till språkstudier. Sedan 16 år lärare i engelska och tyska på ett högstadium i Tranås. Har kvackat i ämnet teknik i c.a 8 år. Började med data först för 2 år sedan genom en kurs på egna skolan. Greps snabbt av iver att lära mig basic-programmering. Insåg det nödvändigt om jag skulle kunna ha någon nytta av datorn för språk. Lärobok i programmering har varit andras program och basic-manualen.

Som tidigare nämnts i denna artikelserie har skolan sedan ett par år obligatorisk dataundervisning. Utöver denna har man på de flesta skolor s.k. datorstöd i olika ämnen, vilket innebär att man t.ex i svenska skriver skisser på ordbehandlare, i matte tränar räknefärdighet, i geografi namngeografi o.s.v.

Självklart borde man även kunna använda datorerna för språk. Tyvärr är detta inte så vanligt, trots att de bör kunna ge utmärkt språkträning.

Varför har då datorerna inte kommit in mer i språkundervisningen? Följande skäl tycker jag mig ha märkt:

1. Dåliga program - ej för egna läromedlet.
2. Rädsla hos lärarna att ej kunna klara av problem.
3. Små resurser för utbildning.
4. Trångt i datasalen.

PROGRAMMEN

Det mesta kring datorer från början kretsar runt matematik och naturvetenskap. Program-makarna har självfallet varit mest intresserade av att göra skol-program inom sitt gebit. Några har totalt ihop språkprogram som t.ex de erbarmligt dåliga GLOSPROG.MWB och GLOSOR.BAS, som följer med MB skolnätverk. Som språklärare förfasar man sig över metodiken.

Ett exempel:

På skärmen visas: Översätt SKRIKA - Eleven svarar: YELL -
Datorn svarar: Fel! SKRIKA heter SCREAM

Sedan går man raskt vidare till nästa uppgift. Där sitter nu eleven förvirrad och undrar om hans svar verkligen var fel.

Sådana program kan inte accepteras. Jag har gjort om MWB-programmet så att det klarar synonymer. Sedan har jag arbetat vidare och lagt in fler finesser. Jag anser att ett bra glosprogram bör ha följande möjligheter:

- Valfritt antal synonymer
- Olika svårighetsnivåer (läraren har klassat glosor som lätta medel eller svåra)
- Olika nivåer för korrekt stavning (eleven kan få rätt trots små stavfel)
- Spärr att gå vidare innan man skrivit ordet rätt.
- Gärna ha ljudeffekter (Ger bättre flyt i träningen)
- För tyska bör det ha tysk-y och dubbel-s

En variant på temat glosträning är program där man parar ihop ett svenskt och ett utländskt ord. Har gjort ett sådant, SMALL.MWB med glosfiler för både engelska och tyska eller egna andra filer. Samma glosfiler används f.ö. också i mitt program MEMORY.MWB.

Att fylla i luckor i text från t.ex dagens läxa är en annan typisk uppgift. Utmärkt för detta är Wirebys WORDS.COM, särskilt om man har läroboksserien FACES, ur vilken han tagit sina texter.

Tyvärr tar de flesta språkprogram upp en massa plats på hårddisken med alla textfiler m.m. Wireby föreslår att man kan köra från diskett på E-enheten. Visst går det bra för den som är van att sköta systemet, men metoden kan knappast rekommenderas för gemene språklärare.

För att träna att skriva löpande text använder jag ett program, TEXTSU.MWB, vilket även det har separata textfiler. Här får eleven se en enda rad av utländska texten en kort stund och ska, när texten försvunnit skriva av vad han sett. En senare utveckling av programmet är att de duktigaste eleverna i stället får se meningen på svenska och ska översätta den.

Ett annat användningsområde är att på ett aptitligt sätt träna grammatik. Do-omskrivning i engelska, tysk adjektiv-och verb-

böjning eller bestämd artikel på substantiv kan göras till ett nöje för eleverna. Extra stimulerande blir det med lite grafik i programmen. Jag har bl.a ett program, ARTIKEL.MWB som tränar tyska der, das eller die (eller vad som helst som kan delas in i tre grupper), där man ska flytta ord på skärmen till olika fack. I programmet ORDF.MWB flyttar man om ord så att man får en mening med rätt ordföljd.

I ett annat av mina program (SNABBENG.MWB och SNABBTY.MWB) visas ordenord med jättestora bokstäver, där varje tecken tar upp 4 matriser (alltså drygt 4 ggr större än vanlig skrift). I programmen LABYENG.MWB och LABYTY.MWB går en gubbe i en labyrint, och man får en glosfråga vid vissa ställen. I programmet LINDANS.MWB får eleven bläddra i ett lexikon och para ihop svenska och engelska ord. Den klassiska leken 'hängning' finns i programmet HANGING.MWB.

Programmen här ovan tar inte så stor plats på hårddisken, eftersom uppgifterna finns i data-satser i programmet.

En fördel man som lärare och hobby-programmerare har är, att man har tillgång till gratis testare av programmen. När man själv testkör sina nya program tror man att de är idiotsäkra. Ofta har det hänt att ett nytt program snabbt har kraschat när 15 elever kör det. Elever har en förmåga att göra osannolika fel. Det blir att gå hem och försöka hitta 'lusen' och göra en ny provkörning med elever. Med tiden skulle nog programmen blivit 100% fungerande, om det inte varit för den klåfingrige programmeraren som ständigt vill göra om och lägga in nya finesser i sina program.

RÄDSLÅ HOS LÄRARNA

Tyvärr är många språklärare inte särskilt apparat-minded. Jag har kolleger som oroar sig för att använda video i klassen, för att de inte tror sig klara av det om något krånglar. Än mer påtaglig är rädslan för datorer. Även om en språklärare gått en intern datakurs och kanske lärt sig lite ordbehandling, så känner han sig knappast mogen att leda datalektioner. Det krävs rutin att klara av fel som kan uppstå, och den får man bara genom att syssla med datorer. Ta en så enkel sak som att en elev får stopp i ett program. Du måste snabbt kunna göra om datorn låst sig och man måste trycka reset, eller om det räcker med att starta om med RUN. Du bör kunna upptäcka om ett fel beror på en enkel glappkontakt. Vi lärare vill kunna det vi sysslar med minst lika bra som eleverna. Även om jag säger till min kollega, att eleverna själva klarar av att starta basic-program, så vill läraren ändå känna att han är lika säker som eleverna.

UTBILDNING

Hur ska man få kollegerna att 'tända' på datorer? Sedan man utbildat första generationen data-lärare och erbjudit internkurs för övriga, anser ledningen att man haft nog med utbildning. För något år sedan fick jag till nåds leda en halvdag med mina språkkolleger, eftersom en inbokad föreläsare fått förhinder. Efter

den dagen var det nära att de andra språklärarna vågat ha egna datalektioner, men det skulle behövts fler studiedagar. Något mer tid lär inte ges, för studiedagarna är för lång tid framåt avsatta för allmänt svammel, som ska passa alla lärare.

På senare tid har på skolan öppnat sig en möjlighet att få igång språklärarna. Speciallärare har ofta sett fördelarna med dataundervisning. De är oftast med den vanlige läraren som kompanjonlärare i klasserna, och de kan då leda lektionerna i datasalen. Den ordinarie läraren kan lugnt vara med och samtidigt själv lära upp sig.

TRANGT I DATASALEN

På min skola med c.a 12 klasser är detta inte något stort problem. Vi har en fast timme i veckan för varje språkblock och samsas om denna, vanligtvis tre lärare, som har lektion samtidigt i årskursen. Det har gått bra hittills, men om intresset sprider sig, får vi nog låta en lärare ta början av lektionen till data och en annan slutet. På större skolor blir det nog svårt för språken att komma åt salen. Den obligatoriska dataundervisningen tar där för stor plats.

HUR OFTA HAR MAN DATALEKTIONER?

Man bör inte ha en övertro på nya inlärningsmetoder. Datorn kan inte lära ut språk, och eleverna ska inte lära sig nya moment på datorer. Först måste man grundligt lärt in sakerna i klassrummet. Datorns roll blir att ersätta monotona skrivövningar i böcker eller på stenciler.

Själv använder jag aldrig mer än högst 20 min åt gången, kanske en gång i veckan eller varannan vecka per klass. Först kör vi seriös träning med t.ex glos-eller textkontroll eller grammatik. Elever som är klara får sedan några minuter med stimulansprogram.

INSTALLATION AV PROGRAM

Som jag sa tidigare, tar språkprogram upp en massa plats på hårddisken. Vi har valt att lägga program med separata glosfiler på enhet D och sådana med datasatser på A. Snart blir det ganska fullt på enhet D, så jag tror jag får börja ha filer som hör till höstterminen på en diskett och bara flytta över dem till hårddisken, när de behövs.

Samtliga av mina program som nämns i artikeln kan man få genom Microbiten. Den som vill veta mer om programmen får gärna skriva eller ringa till mig.

Ragnar Suneson
Februari 3
573 38 Tranås
0140/ 111 52

Det finns två, i princip olika, sätt att se på hur datorn skall användas i skolan. Den ena uppfattningen är, att man enbart skall använda färdiga program, rena tillämpningar. Den andra uppfattningen säger, att vi skall lära eleverna grunderna, hur det fungerar. Själv tycker jag att sanningen, som oftast, ligger någonstans mittemellan. Om eleven vet hur ett enkelt MoS-program (MoS = Mät- och Styr) fungerar, så tror jag att han/hon inser att många "datorstyrda" apparater i vardagslivet inte är så särskilt märkvärdiga. Dessutom så kan han/hon själv göra enkla tillämpningar hemma på sin egen dator. Jag vet att många av mina elever experimenterar själva hemma med MoS. Att kunna grunderna, stimulerar ofta till egna experiment.

Som en jämförelse, är ju dessutom den mesta fysik och kemiundervisningen av grundläggande karaktär: Vi lär eleverna hur ett motstånd fungerar, vad en transistor är osv. Inte låter vi elektronikavsnittet i fysiken enbart vara, att eleverna sitter och skruvar på rattarna på en transistorradio!

Jag tror att man skall börja MoS-undervisningen med att lära ut hur datorn arbetar med in/ut-signalerna. Det blir då nödvändigt att arbeta med binära tal. Eftersom det är ett nytt begrepp, tar det lite tid för eleverna att vänja sig med detta sätt att räkna, men ganska snart blir det enkelt.

Jag brukar låta eleverna skriva in följande program:

```
100 GOSUB 1000
120 D=IN(0)
110 CLS
130 IF D=A THEN 120 ELSE LET A=D
140 CLS:CURS25,7:PRINT A
150 GOTO 120
1000 OUT 1,207
1010 OUT 1,255
1020 RETURN
```

Med hjälp av strömställare, så kortsluter därefter eleven varje "kanal" (eller "inbit") för sig och för in värdet i en tabell:

| | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|
| Värde: | | | | | | | | |
| Inbit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

När tabellen är klar, kan man förklara hur datorn arbetar med binära tal och ASCII-koder (inbitarnas värde).

Efter detta brukar jag lära eleverna, hur datorn kan översätta dessa koder till tecken vi lättare kan förstå. Följande komplettering av rad 140 ovan skriver ASCII-tecknet:

```
140 CLS:CURS25,7:PRINT A,CHR(A)
```

Med hjälp av MoS-kortet kan man sedan ge instruktioner till datorn i form av binära tal (=strömställarna på MoS-kortet). För många elever med egna datorer, kan detta vara en aha-upplevelse: "Är det så här det fungerar!" Aha-upplevelsen kan gälla hexadecimala styrkoder, hur man definierar egna tecken osv. Denna

kunskap tror jag det är viktigt, att vi ger våra elever.

Efter dessa arbetspass, är eleven faktiskt mogen för att göra egna, enkla tillämpningar. Man kan låta olika brytande/slutande komponenter ersätta strömställarnas brytande/slutande funktion. Istället för att använda den traditionella "maskningen", tycker jag att det är bättre att arbeta direkt med värdet på inbitarna. Det är visserligen elegantare att maska, men eleverna begriper ingenting. Och det är ju, för åtminstone varje grundskollärare, en självklarhet att använda så enkla vägar som möjligt.

I programexemplet ovan, kan rad 130 ersättas med:
130 IF D=0 THEN 120

Rad 140 OCH 150 ändras till:
140 CLS:CURS 25,7:PRINT "LARM!":PLAY 10,8
150 GOTO 110

Nu har man ett larm som känner av om något bryter sensorn, som är kopplad till någon ingång. Sensorn kan vara ett enkelt fotomotstånd, en magnetisk brytare kopplad till en dörr, osv.

Det är alltså mycket lätt att använda microbee-datorn på detta sätt. Eleverna förstår också principen snabbt och det är ett populärt avsnitt.

Att fortsätta och göra mer avancerade mät-kopplingar är både stimulerande och lätt för eleverna. Innan man gör detta, kan det kanske först vara lämpligt att också lära eleven hur man kan styra med datorn.

Om eleven lärt sig dessa första enkla sätt att mäta (binära tal och deras decimala värde), blir det lätt att vända processen och istället skicka ut signaler från datorn.

Följande program klarar det:

```
0100 GOSUB 1100
0110 CLS:PRINT"Vilken diod vill du tända?"
0120 INPUT"Ange värdet: ";D
0130 OUT 0,D
0140 GOTO 110
1100 OUT 1,207
1110 OUT 1,0
1120 RETURN
```

Med detta enkla program kan man tända en eller flera dioder (eller öppna utgångarna)

Med dessa båda metoder för att låta datorn ta emot och sända ut signaler, kan man göra en syntes: låta datorn ta emot på kanal 0-3, och sända ut på kanal 4-7.

Subrutinen 1200 klara av detta:

```
1200 OUT 1,207
1210 OUT 1,15
1220 RETURN
```

Nu kan datorn ta emot fyra olika signaler, bearbeta dem beroende på deras värde (dvs vilka sensorer som aktiverats), och därefter sända ut en eller flera signaler.

Som sensorer finns en uppsjö av möjligheter, Clas Ohlson-katalogen har massor av lämpliga sensorer: IR-sensorer som klarar upp mot 10 meter, ultraljud, magnetiska brytare, fotomotstånd, termistorer osv. Kopplar man då dessutom ett reglerbart motstånd i serie med fotomotståndet eller termistorn, så kan man göra en krets som slår till vid ett visst ljus (och t.ex. tänder en lampa) eller vid en viss temperatur (och då startar en fläkt).

Till utsignalerna kan också kopplas massor av saker, endast plånboken och fantasin begränsar.

Förutom att eleverna själva kan göra MoS-program, så är det lätt att som lärare göra ett antal färdiga tillämpningsprogram. Många realistiska tillämpningar kan göras. I det paket som jag konstruerat, finns program för att mäta den mänskliga reaktionsförmågan på såväl ljud- som ljussignal, accelerationen hos en rullande kula på ett lutande plan, svängningstiden för en pendel, olika tids- och hastighetsmättningsprogram, osv. I dessa tillämpningsprogram kan man då också låta datorn göra komplicerade matematiska beräkningar, som t.ex. medelvärde och varians osv.

Som väl läsaren själv insett vid det här laget, så är det faktiskt mycket enkelt att MoS:a med microbee. Endast enkla instruktioner behövs, har man ett vettigt MoS:kort är det helt ofarligt och det är ROLIGT!

I nästa avsnitt tänker jag visa hur man kan MoS:a med microbee:s serieport. Jag kommer då också att referera till en artikel och några program som jag fått mig tillsänt av Staffan Tymark på Bleketskolan i Rönnäng.

Lycka till med att MoS:a på microbee och hör gärna av Dig med tips, frågor, artiklar osv. Vår tidning skapas av de som inte bara tar emot, utan också ger ut egna tips, tankar och ideer! Hör av Dig.

Hälsningar
Bo Hallberg
Kapellgatan 7
570 20 BODAFORS
Tel: 0380-307 46

Eftersom ett flertal personer har hört av sig om olika problem vid överföringar av texter mellan Microbee och PC maskiner, så tyckte jag det var dags att reda ut begreppen lite.

För att föra över program mellan dessa två system så kan man gå till väga på två olika sätt. Ett sätt är att koppla ihop de två maskinernas serieportar med en kabel eller via modem och sedan föra över filerna med två terminalprogram. Alternativ två är att låta Microbeeen läsa/skriva PC disketter med programmet PCBEE.

Det sista alternativet är det vanligaste och fungerar också utmärkt om man har klart för sig att PC maskiner använder en annan teckenuppsättning än vad som är standard på de flesta datorer. Detta påverkar förstas överföringar via serieportarna också. Den vanligaste teckenuppsättningen, som används på Microbee, kallas ASCII. I ASCII finns det 128 olika tecken som återges genom att använda 7 av de 8 bitar som ingår i en byte. Eftersom IBM självklart inte nöjde sig med att bara kunna återge 128 olika tecken så hittade de på en egen variant av ASCII. Denna uppsättning innehåller 256 tecken och använder sig av alla de 8 bitar som en byte innehåller. En viktig skillnad mellan dessa två system är att våra unika svenska bokstäver ÅÄÖÅSÖ har fått olika koder och därför måste man översätta dem när man utbyter texter mellan Microbee och PC. Detta låter sig göras i programmet PCBEE där man kan specificera om man vill ha åäö konvertering eller ej. (Se också programmet CPM-DOS under rubriken 'Kort om nya program' ovan.)

Problemet är att det är mer krångligt än så här. Eftersom den vanliga ASCII uppsättningen bara använder 128 tecken utav 256 möjliga så har man kommit på att man kan använda de resterande till att lägga in speciella kontrollkoder i texter. WordStar utnyttjar den senare metoden för att till exempel särskilja de extra mellanslag som sätts in i texten för att få rak högermarginal. Om man tittar på en text skriven i WordStar med hjälp av TRANSFERS funktion 'Visa' med 'Ta med bit 8' funktionen aktiv så kommer man se en hel del tecken som är i invers. När man sedan för över denna text till PC:n så kommer den tolka dessa inverterade tecken (som har 8e biten satt) som andra bokstäver eller tecken i dess utökade teckenuppsättning. På något sätt så måste man alltså bli av med dessa kontrolltecken.

Nu är det dags att plocka fram det (illa) beryktade CP/M programmet PIP. PIP skall finnas på din master diskett som du fick med när du köpte datorn. Detta helt otroliga allt-i-allo kopieringsprogram kan nämligen nollställa den 8e biten på alla tecken och på det viset rensa texten ifrån de störande tecknen. Om vi kallar den text som vi vill rensa för WS.TXT och den rensade texten för RENSAD.TXT så skall man skriva så här direkt i SHELL eller CCP: PIP RENSAD.TXT=WS.TXT&A. De parametrar som står mellan &A bestämmer hur kopieringen skall gå till. Observera att man måste använda stora Å och Ä. För mer information om dem se under rubriken PIP i Microbee disk system manual. PIP kan nämligen mycket mer än det som visades i detta exempel.

Daniel Grönjörd

Efter ett tips ifrån en av klubbens medlemmar, vars namn tyvärr har försvunnit någonstans i redaktionens pappershögar, så har det framkommit att det snarare rör sig om skrivfel i program listningen än att den sista delen av programmet skulle saknas.

Förutom de fel som påtalades i inledningen till program listningen så har de följande också uppmärksammats. Adresserna nedan hänför sig till de som står vid den vänstra kanten av den ursprungliga program listningen.

- Hexadecimala tal som börjar med en bokstav måste ha en nolla först. Detta beror på att assemblern måste kunna skilja på hexadecimala tal och labels (som därför aldrig får börja med en siffra).
- På adress 1000 skall det stå PCGIN. Bidesstrecket skall alltså inte finnas med.
- På adress 101A skall det stå CALL PRTMEN istället för CALL PRT.
- Det skall vara ett komma mellan NZ och NRKLOP på adress 1032. Detta fel förekommer på flera ställen och skall åtgärdas på samma sätt.
- Det står i kommentaren på adress 106F att programmet skall hoppa till SUDLOP så det skall stå SUDLOP där istället för NYBLOP.
- På adress 1080 fattas ena delen av parantesen. Det skall förstås stå LD (HL),D.
- På adress 10F1 måste det stå JP NZ,KRASCH. Annars kommer flygplanet krascha varenda gång det försöker ge sig iväg.
- På adress 1101 fattas registret HL. Det skall stå LD (12FA),HL.
- På adress 113B kan man skriva PGMLOP istället för den adress som står där.

Förhoppningsvis skall dessa rättelser få programmet att fungera. Om någon har hittat några andra fel eller har några förbättringar så hör gärna av er till redaktionen.

PROGRAMMERING

Tanken med den här artikeln är att visa hur ett program som utför en viss procedur kan se ut i olika programspråk. Utifrån de resultat som dessa program uppvisar så skall jag försöka urskilja några programspråk som lämpar sig bättre än de andra för just denna procedur. De programspråk vi skall titta på är: Ada, Basic, C, Comal, Forth, Fortran, Pascal och PL/I.

Programmet som skall testas i de olika programspråken räknar ut printal och kallas ibland för ett 'sieve' program. Detta prövar framför allt den matematiska styrkan i ett programspråk och därför har sådana språk som Cobol och Prolog inte tagits med. Programmet är dock ett rättså generellt styrkeprov på de olika

programspråken eftersom beräkningshastigheten är avgörande i nästan alla program. För att ge alla medlemmar i klubben en chans att förstå vad de olika programmen utför och vad de olika testvärdena innebär, så har jag gjort ett motsvarande program i MicroWorld Basic. Alla tester är utförda på en helt vanlig Microbee 128 med två diskettenheter. M-driven har inte använts utan all kompilering har skett med kompilatorn på diskettenhet A: och programtexten på diskettenhet B:.

Nedan följer en lista med de olika programspråken, versionsnummer och tillverkarnas namn.

| Språk | Version | Tillverkare |
|--------------------------|---------|--|
| Aztec C | 1.05c | Manx Software Systems |
| BD Software C | 1.50a | BD Software |
| Comal-80 | 1.1 | Metanic |
| Forth-83 | 1.0.0 | Public Domain |
| Fortran-80 | 3.44 | Microsoft |
| Microsoft Basic | 5.21 | Microsoft |
| Microsoft Basic Compiler | 5.3 | Microsoft |
| MicroWorld Basic | 6.23s | MicroWorld |
| PL/I-80 | 1.3 | Digital Research |
| Small-C | 1.4 | John Hill (efter Ron Cain) |
| Supersoft Ada | 2.10 | Supersoft och Maranatha Software Systems |
| Turbo Pascal | 3.01A | Borland |
| ZBasic | 3.10 | Zedcor . |

För dem som bara har programmerat Basic kan termen kompilator kanske vara okänd, så jag skall försöka förklara vad skillnaden mellan en interpretator och en kompilator är.

Interpretatorn kontrollerar och utför varje instruktion innan den fortsätter till nästa. Interpretatorn vet aldrig vilken nästa instruktion är. Radnummer eller "labels" är nödvändiga för att interpretatorn skall kunna veta vart den skall dirigera programflödet när den stöter på ett GOTO eller någon annan instruktion som ändrar programflödet. Fördelarna med interpretatorer är att de är lätta att lära sig och att det är lätt att göra ändringar i programmen. Nackdelarna är att de är långsamma och att de använder datorns minne på ett ineffektivt sätt. Programmen skrivs vanligen i en rad-editor som finns i interpretatorn. De interpreterande språken är Comal-80, Microsoft Basic och MicroWorld Basic.

Ett program som har skrivits i ett kompilerande programspråk "vet" vad som kommer hända eftersom kompilatorn redan har gått igenom programmet flera gånger och kontrollerat allt vid kompileringen. Programmen skrivs i en editor eller ordbehandlare och kompileras sedan med en kompilator. Fördelarna med kompilerande programspråk är att programmen blir snabba och att programmet blir till en separat maskinspråksfil (.COM-fil) på disketten. De kompilerande språken är Aztec C, BD Software C, Forth-83, Fortran-80, PL/I-80, Small-C, Supersoft Ada, Turbo Pascal och ZBasic. Forth-83 kompilerar på ett ackumulerande sätt dvs. varje programrad kompileras direkt och läggs till programspråkets repertoar av kommandon.

För att krångla till det hela så finns det programspråk som är

interpreterande men där programmen ändå går att kompilera. Man kan alltså utveckla ett program i en vanlig interpretator och sedan kompilera dem med en separat kompilator. Ett sådant språk är kompilerad Microsoft Basic (har namnet Microsoft Basic Compiler i tabellerna nedan).

Först skall vi titta på de olika programmen som har testats. Vi börjar för enkelhetens skull med den vanliga MicroWorld Basic.

En begränsning som genast märker med MicroWorld Basic är att man inte kan ha längre variabelnamn än två tecken. Instruktioner som WHILE och WEND saknas också vilket gör att man måste ta till GOTO satser istället. Basic står för övrigt för Beginners All-purpose Symbolic Instruction Code.

```
00100 S=8190
00110 DIM F(S)
00120 PRINT"10 iterationer -";
00130 FOR R=1 TO 10
00140     C=0
00150     FOR I=0 TO S
00160         F(I)=1
00170     NEXT I
00180     FOR I=0 TO S
00190         IF F(I)=0 THEN 270
00200         P=I+I+3
00210         K=I+P
00220         IF K>S THEN 260
00230             F(K)=0
00240             K=K+P
00250         GOTO 220
00260         C=C+1
00270     NEXT I
00280 NEXT R
00290 PRINT C;" primtal."
00300 END
```

Det som är unikt med MicroWorld Basic är förstås att man har tillgång till Microbees grafik på ett enkelt sätt.

Nästa program är det som användes vid testningen av Microsoft Basic, Microsoft Basic Compiler och ZBasic. Eftersom dessa tre språk är så lika varandra (Microsoft Basic och Microsoft Basic Compiler använder exakt samma instruktioner) så kunde jag använda samma program till dem.

```
100 SIZE%=8190
110 DIM FLAGS%(8190)
120 PRINT"10 iterationer -";
130 FOR ITER%=1 TO 10
140     COUNT%=0
150     FOR I%=0 TO SIZE%
160         FLAGS%(I%)=1
170     NEXT I%
180     FOR I%=0 TO SIZE%
190         IF FLAGS%(I%)=0 THEN 270
```

```

200     PRIME%=I%+I%+3
210     K%=I%+PRIME%
220     WHILE K%<=SIZE%
230         FLAGS%(K%)=0
240         K%=K%+PRIME%
250     WEND
260     COUNT%=COUNT%+1
270     NEXT I%
280 NEXT ITER%
290 PRINT COUNT%; "primal."
300 END

```

Som man kan se så innehåller dessa språk ett par instruktioner som underlättar programmeringen betydligt. Det är WHILE och WEND instruktionerna som ger det mer lättlästa programmet. Här har alla GOTO instruktioner försvunnit. Längre variabelnamn kan också användas. Procenttecknen som förekommer efter varje variabel betyder att den variabeln är en heltals-variabel.

Nästa programspråk är Comal-80. Comal står för COMMON Algorithmic Language och är avsett främst för undervisning. Det används nog mest på Compis datorerna. Det är ett sorts mellanting mellan Basic och Pascal.

```

0100 SIZ#:=8190
0110 DIM FLAGS#(0:SIZ#)
0120 PRINT "10 iterationer - ";
0130 FOR ITER#:=1 TO 10 DO
0140     COUNT#:=0
0150     FOR I#:=0 TO SIZ# DO
0160         FLAGS%(I#):=1
0170     NEXT I#
0180     FOR I#:=0 TO SIZ# DO
0190         IF FLAGS%(I#) THEN
0200             PRIME#:=I#+I#+3
0210             K#:=I#+PRIME#
0220             WHILE K#<=SIZ# DO
0230                 FLAGS%(K#):=0
0240                 K#:=K#+PRIME#
0250             ENDWHILE
0260             COUNT#:=COUNT#+1
0270         ENDIF
0280     NEXT I#
0290 NEXT ITER#
0300 PRINT COUNT#; "primal."

```

Det första högnivå språket (dvs. ej assembler) var Fortran. Det kom redan på 50-talet. Fortran står för FORMula TRANslator. Som man direkt ser så har Fortran inspirerat utvecklingen av Basic. Bla. så finns här en hel del GOTO instruktioner. Radnummer behövs dock bara på de platser som andra instruktioner i programmet hänvisar till. Fortran används framför allt till vetenskapliga tillämpningar och bör undvikas i alla andra sammanhang.

PROGRAM SIEVE

```
LOGICAL FLAGS(8190)
INTEGER SIZE,I,PRIME,K,COUNT,ITER

SIZE = 8190

WRITE(1,100)

DO 60 ITER = 1,10
  COUNT = 0

  DO 20 I = 1,8190
    FLAGS(I) = .TRUE.

  DO 50 I = 1,8190
    IF (.NOT. FLAGS(I)) GO TO 50
    PRIME = I + I + 3
    K = I + PRIME
  30   IF (K .GT. SIZE) GO TO 40
      FLAGS(K) = .FALSE.
      K = K + PRIME
      GO TO 30
  40   COUNT = COUNT + 1
  50   CONTINUE

  60   CONTINUE

  WRITE(1,110) COUNT
  STOP SIEVE

100   FORMAT('*10 iterationer - ')
110   FORMAT('* ',I4,' primtal.',/)
```

END

Fortran använder ett väldigt annorlunda system för utskrift på bildskärmen. Man anger i WRITE-satsen ett radnummer som pekar på en FORMAT sats där texten finns.

Forth liknar inget annat programspråk och har en mycket kompakt programtext. Det används i första hand i styr- och regler-sammanhang. Forth är ett sorts operativsystem samtidigt som det är ett programspråk och alla program måste köras ifrån Forth. Forth-83 innehåller dessutom en assembler (8080), debugg-hjälpmiddel och källkod för multitasking. Forth-83 är skrivet i Forth-83!

```
8190 CONSTANT SIZE
0 VARIABLE FLAGS SIZE ALLOT
```

```
: SIEVE
  ." 1 iteration - "
  FLAGS SIZE 1 FILL
  0 SIZE 0
  DO FLAGS I + C@
```

```

IF I DUP + 3 + DUP I +
BEGIN DUP SIZE <
  WHILE 0 OVER FLAGS + C! OVER + REPEAT
  DROP DROP 1+
THEN
LOOP
. ." primal." ;

```

Observera att detta program endast utför en iteration och den resulterande exekveringstiden alltså måste multipliceras med tio för att kunna jämföras med de andra programmen. Programmet använder sig bara av två variabler, alla andra beräkningar sker på stacken. Om man vill lära sig att programmera i Forth så skall man läsa böckerna Starting Forth och Thinking Forth som är mycket bra.

C är ett programspråk som är på modet. Ändå är det inte särskilt nytt. C kännetecknas av sin flexibilitet och av att vara systemnära. Detta program är skrivet i Small-C. Som man hör på namnet så är detta inte en särskilt komplett C-kompilator. Small-C skrevs ursprungligen av Ron Cain och det här är en förbättrad version. Small-C är skriven i Small-C.

```

#include CRUN.LIB
#include CONIO.LIB
#include NUMIO.LIB

#define TRUE 1
#define FALSE 0
#define SIZE 8190
#define SIZEPL 8191

char flags[SIZEPL];
int i,prime,k,count,iter;

main() {
  puts("10 iterationer - ");

  iter = 1;
  while (iter <= 10) {
    count = 0;

    i = 0;
    while (i <= SIZE) {
      flags[i]=TRUE;
      i++;
    }

    i = 0;
    while (i <= SIZE) {
      if (!flags[i]) {
        prime = i + i + 3;
        k = i + prime;
        while (k <= SIZE) {
          flags[k] = FALSE;
          k = k + prime;
        }
      }
      i++;
    }
    count++;
  }
}

```

```

        )
        count++;
    )
    i++;
)
    iter++;
)
putdec(count);
puts(" primtal.");
crlf();
}

```

För den som vill prova på C så är Small-C en utmärkt kompilator.

Följande C program användes för att testa både BD Software C och Aztec C. Aztec C finns till många olika datorer och det är en fullständig C kompilator med t.ex. flyttal och register variabler. BD Software C är en traditionell C kompilator för CP/M datorer. Den ger mycket snabba och kompakta program och lämpar sig för all sorts programmering som inte har med flyttal att göra. Den största delen av de program som finns i klubbens bibliotek av Public Domain program är skrivna i BD Software C.

```

#define TRUE 1
#define FALSE 0
#define SIZE 8190
#define SIZEPL 8191

char flags[SIZEPL];
int i,prime,k,count,iter;

main() {
    printf("10 iterationer - ");

    for (iter = 1; iter <= 10; iter++) {
        count = 0;

        for (i = 0; i <= SIZE; i++) flags[i]=TRUE;

        for (i = 0; i <= SIZE; i++) {
            if (flags[i]) {
                prime = i + i + 3;
                k = i + prime;
                while (k <= SIZE) {
                    flags[k] = FALSE;
                    k+=prime;
                }
                count++;
            }
        }

    }

    printf("%d primtal.\n",count);
}

```

För den som inte är van vid C så kan det här vara svårt att tyda. Om man t.ex. tittar på den sista 'printf' satsen så ser den mycket komplicerad ut. "%d" representerar det tal som "count" innehåller och bestämmer alltså på vilken plats talet skall skrivas ut på. "\n" betyder att en radframmatning skall ske på bildskärmen. Lägg märke till sådana instruktioner som "count++" och "k+=prime" som skulle skrivas "count=count+1" respektive "k=k+prime" i Basic.

PL/I (Programming Language One) är IBMs stordator språk. Den här PL/I kompilatorn är gjort av samma företag som har gjort CP/M dvs. Digital Research. PL/I är ett mycket stort språk med ett mängder av olika instruktioner. Den som har programmerat i C eller Pascal bör dock kunna känna igen sig en del. Många av instruktionerna går att förkorta, t.ex "procedure" går att förkorta till "proc" om man vill. PL/I är också ett fritt språk i det avseendet att man kan skriva alla procedurer och deklARATIONER huller om buller i programmet utan att kompilatorn blir konfunderad. De deklARATIONER som står i början av det här programmet kunde t.ex. lika gärna stått sist.

```
sieve:
  procedure options(main);

  %replace
    size by 8190,
    true by '1'b,
    false by '0'b,
    bel by 7;

  dcl flags(0 : size) bit(1);
  dcl (i,prime,k,count,iter) fixed;

  put list('10 iterationer - ');

  do iter = 1 to 10;
    count = 0;

    do i = 0 to size;
      flags(i) = true;
    end;

    do i = 0 to size;
      if flags(i) = true then do;
        prime = i + i + 3;
        k = i + prime;
        do while (k <= size);
          flags(k) = false;
          k = k + prime;
        end;
        count = count + 1;
      end;
    end;

  end;
  put list(count,'primal.');
```



```

    put skip;
end sieve;

```

Eftersom PL/I-80 var det första mikrodator språket som kunde hantera overlayer på ett enkelt sätt (dvs. program som är större än minnet i datorn) så skrevs många av de första avancerade programmen i PL/I-80. PL/I har också en mycket avancerad filhantering som lämpar sig för hantering av databaser (register), det finns t.ex. inbyggd indexering precis som i DBASE.

Det populäraste kompilerande programspråket på Microbee är säkert Turbo Pascal. Turbo Pascal är mycket lättanvänt och ger också snabba program. Pascal program är (oftast) väldigt lättlästa.

```
PROGRAM Sieve;
```

```
CONST
```

```
    size = 8190;
```

```
VAR
```

```

    flags : ARRAY(0..size.) OF BOOLEAN;
    i      : INTEGER;
    prime  : INTEGER;
    k      : INTEGER;
    count  : INTEGER;
    iter   : INTEGER;

```

```
BEGIN (* Main *)
```

```
    Write('10 iterationer - ');
```

```
    FOR iter:=1 TO 10 DO
```

```
        BEGIN
```

```
            count:=0;
```

```
            FOR i:=0 TO size DO flags(i.):=TRUE;
```

```
            FOR i:=0 TO size DO
```

```
                BEGIN
```

```
                    IF flags(i.) THEN
```

```
                        BEGIN
```

```
                            prime:=i+i+3;
```

```
                            k:=i+prime;
```

```
                            WHILE k<=size DO
```

```
                                BEGIN
```

```
                                    flags(k.):=FALSE;
```

```
                                    k:=k+prime;
```

```
                                END (* WHILE *);
```

```
                                count:=Succ(count);
```

```
                            END (* IF *);
```

```
                    END (* FOR *);
```

```
            END (* FOR *);
```

```
            WriteLn(count, ' primtal.');
```

```
        END (* Sieve *).
```

I Turbo Pascal blir programmen ofta mycket prydliga. Ta en sådan sats som "count:=Succ(count)" som motsvarar "count=count+1" i Basic.

Som sista programspråk kommer Supersoft Ada. Det är mycket influerat av Pascal och är också väldigt lättläst. Tyvärr blir alla program långsamma om man jämför med andra kompilerande språk.

```
procedure SIEVE is
  pragma OPTIMIZE(TIME);

  SIZE : constant INTEGER := 8190;
  FLAGS : array (0 .. SIZE) of BOOLEAN;
  I      : constant INTEGER;
  PRIME : INTEGER;
  K      : INTEGER;
  COUNT : INTEGER;
  ITER  : constant INTEGER;

  for I      use at 128;
  for PRIME use at 130;
  for K      use at 132;
  for COUNT use at 134;
  for ITER  use at 136;
  for SIZE  use at 138;
  for FLAGS use at 20000;

begin
  PUT ("10 iterationer - ");

  for ITER in 1..10 loop
    COUNT := 0;

    for I in 0..SIZE loop
      FLAGS(I) := TRUE;
    end loop;

    for I in 0..SIZE loop
      if FLAGS(I) then
        PRIME := I + I + 3;
        K := I + PRIME;
        while K <= SIZE loop
          FLAGS(K) := FALSE;
          K := K + PRIME;
        end loop;
        COUNT := COUNT + 1;
      end if;
    end loop;

    end loop;
  PUT (COUNT);
  PUT (" prmtal.");
  NEW_LINE;
end SIEVE;
```

Nu skall vi titta på de resultat som dessa testprogram gav. För att ta reda på hur snabba de olika språken är så har jag mätt hur lång tid programmen har tagit för att utföra alla beräkningar. Observera att tiderna för Comal-80, Microsoft Basic och MicroWorld Basic är uppskattningar baserade på en test som genomför programmet på exakt en tiondel av tiden. En enkel multiplikation har sedan givit resultatet i tabellen. Jag har inte suttit och väntat i över en halv timme med klockan i handen på att dessa program skall bli klara!!

En program skrivet direkt i assembler skulle uppskattningsvis ha tagit drygt 8 sekunder för att utföra motsvarande beräkningar.

| Språk | Exekveringstid i sekunder |
|--------------------------|---------------------------|
| PL/I-80 | 17 S |
| BD Software C | 18 S |
| Fortran-80 | 20 S |
| Microsoft Basic Compiler | 23 S |
| Aztec C | 26 S |
| Turbo Pascal | 27 S |
| ZBasic | 29 S |
| Small-C | 37 S |
| Forth-83 | 93 S |
| Supersoft Ada | 101 S |
| Comal-80 | 1850 S |
| Microsoft Basic | 1870 S |
| MicroWorld Basic | 2370 S |

Här kan man tydligt se hur stor skillnad det är i exekverings-hastighet mellan interpreterande och kompilerande programspråk. PL/I-80 är ungefär 140 gånger snabbare än MicroWorld Basic men bara något snabbare än BD Software C. Fortran-80, som är det traditionella beräkningsspråket, hade med stor sannolikhet varit snabbast om det hade gällt noggrannare beräkningar. Något som är anmärkningsvärt är att kompilerad Microsoft Basic till och med är snabbare än Turbo Pascal. Att Ada är det långsammaste kompilerande språket kommer dock inte som någon överraskning eftersom det har varit ett av huvudargumenten mot det språket. Trots det så har Ada blivit officiell standard både i USA och inom EG.

Något som CP/M programmeraren måste tänka på är hur stora programmen blir. Med maximalt 64 Kb minne i datorn så blir man tvingad att ha små och kompakta program. Man skall dock inte se detta som någon större begränsning. Att programmen blir stora nu för tiden beror främst på de grafiska interface som används. Men med ett effektivt användande av overlay och chain filer så kan man också göra mycket stora program under CP/M. I denna tabell finns förstås inte de interpreterande språken eller Forth-83 med utan bara de som ger maskinspråks filer.

Språk

Objekt kodens storlek i bytes

| | |
|--------------------------------------|---------|
| Small-C | 2048 b |
| BD Software C | 3712 b |
| Supersoft Ada | 7296 b |
| Turbo Pascal | 8576 b |
| ZBasic | 10368 b |
| PL/I-80 | 14976 b |
| Fortran-80 | 15104 b |
| Microsoft Basic Compiler (15488+896) | 16384 b |
| Aztec C | 16896 b |
| Microsoft Basic Compiler | 25472 b |

I tabellen förekommer kompilerad Microsoft Basic två gånger. Det beror på att man kan låta det så kallade runtime biblioteket ligga som en separat fil på disketten eller slå ihop det med programmet. Att ha runtime biblioteket som en separat fil är att föredra då man har flera program gjorda i kompilerad Microsoft Basic på samma diskett, då dessa kan dela på samma biblioteksfil.

Small-C och BD Software C ger de absolut minsta programmen. Att Small-C programmet är mindre än BD Software C programmet beror på att funktionen Puts användes i Small-C programmet istället för den mycket utrymneskrävande Printf funktionen. Programmet som är skrivet i Small-C är mindre än en tredjedel så stort som motsvarande Turbo Pascal program.

Aztec C tycks ge mycket stora program när man tittar i tabellen ovan. Men ändrar man en enda rad i programmet så halveras storleken. Man flyttar raden "char flags;SIZEPLA;" innanför "Main()" och gör därmed denna array lokal för programmet och oinitierad. Programmet blir istället 8704 bytes långt och kompileringstiden blir 63 S. Exekveringstiden höjs dock till 28 S eftersom lokala variabler är långsammare än globala.

När man testar PL/I-80 uppstår samma problem men tyvärr så har jag inte kommit på hur man deklarerar en array i PL/I så att den inte blir initierad. För att få en uppfattning av hur stort själva programmet är så kan man prova med en array som bara har ett element. Programmet blir då 6784 bytes långt.

När man har med kompilerande språk att göra så finner man att programutvecklingen av stora program ofta berörs av den tid det tar att kompilera programmen, dvs. den tid det tar att omvandla programtexten till ett körbart program. Här finns det ett problem vid bedömningen av hur snabbt språket kompilerar. De traditionella språken låter programmeraren skriva programmen i en separat editor och sköta kompileringen direkt ifrån system nivå (CCP eller SHELL). Men i en del nya språk sköts kompileringen ifrån en kombinerad editor och kompilator. Turbo Pascal och ZBasic hör till den senare kategorin. Här måste man alltså räkna med den tid det tar att ladda in denna editor/kompilator för att kunna jämföra dem med de traditionella kompilerande språken.

Även denna gång så har de interpreterande språken och Forth-83 utslutits eftersom de inte arbetar på det här sättet.

| Språk | Kompileringstid i sekunder |
|--------------------------|----------------------------|
| Turbo Pascal | (5+6) 11 S |
| ZBasic | (7+5) 12 S |
| BD Software C | 22 S |
| Supersoft Ada | 31 S |
| Fortran-80 | 50 S |
| Aztec C | 65 S |
| Microsoft Basic Compiler | 73 S |
| PL/I-80 | 79 S |
| Microsoft Basic Compiler | 95 S |
| Small-C | 165 S |

Aterigen dyker kompilerad Microsoft Basic Compiler upp i tabellen två gånger och det av samma orsak som förra gången. Det går snabbare att kompilera programmet om man väljer att lägga runtime biblioteket separat på disketten. Turbo Pascal och ZBasic är överlägset snabbast på att kompilera programmen. Turbo Pascal kompilerar ett program på bara 48% av den tid som BD Software C tar för att kompilera sitt program på, fastän denna C kompilator är känd för att vara snabb. Det finns dock fördelar med det traditionella systemet. Programmen kan sättas ihop med separata maskinspråksfiler och man får också snabbare reda på alla stavfel etc. som kan förekomma i programmen. Ett sätt att snabba upp kompileringen av programmen på är att kompilera programtexten på M-driven. Det går bra så länge det inte rör sig om mycket stora program. Kompilerings-tiden minskar då till bara 50-70% av den som anges i tabellen beroende på vilket språk man använder.

Jag skall nu försöka sammanfatta dessa test resultat. Om man studerar listorna lite närmare finner man (med lite god vilja) att de fyra främsta programspråken i varje test innehar en särställning jämfört med de andra. Men det är bara tre av dessa programspråk som har godkända resultat i alla tre testerna: BD Software C ger mycket snabba och kompakta program och har en hyfsat kort kompileringstid. Turbo Pascal ger snabba, ganska kompakta program och har den snabbaste kompileringstiden. ZBasic ger ganska snabba program, kompilerar snabbt men ger ganska stora program.

Om det är några programspråk som kan konkurrera med de här tre så är det PL/I-80, Fortran-80 eller Aztec C. Både PL/I-80 och Fortran-80 är fantastiskt snabba när det gäller komplicerade beräkningar. Om det däremot gäller beräkningar med stor decimal noggrannhet så är Fortran-80 att föredra. PL/I-80s styrka är istället snabb heten och den stora mängden av kommandon. Nästan alla de funktioner som man brukar hitta på en avancerad teknisk miniräknare finns inbyggda. Aztec C är en C kompilator som har ett helt Unix kompatibelt bibliotek av funktioner vilket gör den intressant för den som arbetar på olika sorters datorer.

Supersoft Ada ger alldeles får långsamma program för att kunna vara konkurrenskraftig. Det är dock en rättså snabb kompilator som ger kompakta program. Den största nackdelen med Supersoft Ada är att den inte är en komplett implementation av Ada. Det saknas alltså en del instruktioner om man jämför med de stora Ada kompilatorerna. Vill man däremot lära sig Ada så är Supersoft Ada ett förnämligt sätt eftersom kompilatorn är den mest lättanvända med

undantag av Turbo Pascal.

För dem som inte tänker använda Microbee i all evighet utan tänker skaffa sig en annan dator så kan det vara intressant att veta vilka av dessa programspråk som finns på andra datorer. Om man inte kräver att program skall kunna flyttas mellan olika datorer utan några som helst ändringar (vilket man sällan lyckas med när det rör sig om olika operativsystem och maskinvara) så kan man välja nästan vilket programspråk som helst. För att få så lite problem som möjligt bör man välja Turbo Pascal eller ZBasic som båda finns på PC och MacIntosh. PL/I-80, som ursprungligen är ett stordatorspråk, finns på många mini och stordatorer bla. på Digital Equipments Vax maskiner. PL/I-80 finns också till PC under namnet PL/I-86. Programspråk som Forth-83, Fortran-80 och Supersoft Ada följer mer eller mindre de internationella standarder som finns men de har alla sina egenheter som kan skapa problem på den andra datorn. Större C-kompilatorer, som t.ex. BD Software C eller Aztec C, följer till största delen de standarder som finns, vilket gör att man kan flytta programmen utan några större förändringar till t.ex. Unix maskiner. Aztec C finns dessutom till PC, MacIntosh, Atari ST och Commodore Amiga.

När man skall välja vilket programspråk man skall använda så är priset en nog så viktig faktor. Den här tabellen visar vad de olika programspråken kostar. De som finns med när man köper 128an (Microsoft Basic och MicroWorld Basic) och Comal-80 som jag inte har hittat något aktuellt pris på, finns inte med.

Forth-83 är ett Public Domain program som kan beställas från klubben på disketten SIG/M.154. John Hills Small-C kommer finnas på Public Domain disketten MUG.002 tillsammans med en del andra användbara program. Det blir den andra Public Domain disketten med program speciellt utvalda för klubben.

Priserna har hämtats ur Jet Computers Dator Handbok för 1988/89 och Grey Matters prislista. Turbo Pascal och ZBasic säljs även av Network.

| Språk | Pris |
|------------------------------|----------|
| Aztec C (version 1.06d) | 110 Pund |
| BD Software C (version 1.60) | 65 Pund |
| Fortran-80 | 7200 Kr |
| Microsoft Basic Compiler | 4800 Kr |
| PL/I-80 | 5400 Kr |
| Supersoft Ada | 3800 Kr |
| Turbo Pascal 3.01A (engelsk) | 510 Kr |
| ZBasic | 995 Kr |

Som man kan se så är det inte några billiga saker, men som tur är så är de tre intressantaste språken också de billigaste.

BD Software C och Aztec C säljs i England av Grey Matter, 4 Prigg Meadow, Ashburton, Devon TQ 13 7DF. Tel: (0364) 53 499. Grey Matter brukar ha annonser i tidningen Personal Computer (och även i Januari numret av Byte tror jag) där alla programspråken är uppräknade.

Daniel Grönjörd

MICROBEE 128 - PROGRAMVARA - MATRISSKIVARE SAKATA SP-1500

Program på 16 disketter i läsbar diskettbox ingår. Förutom grundpaketet med bl.a. WordBee, WordStar, Databas och Telcom medföljer Microsoft Basic samt mängder av nytto- och spelprogram. Pris: 7000 Kr.

Ring: Jonathan Martinsson 0301-433 51
eller Arne Johnson 0301-433 43

K Ö P E S

För några kollegors räkning köper jag gärna en begagnad Microbee, eventuellt flera. Den skall användas för ordbehandling på främmande språk (t.ex. ryska), varför det är en fördel om Graphic WordBee ingår i programvaran så att vi lätt kan lägga in våra kyrilliska och andra bokstäver. Inga andra krav på finesser, och ju billigare desto bättre.

Ingrid Maier
Döbelnsgatan 24 C
752 37 Uppsala

Tel hem: 018-14 52 34 (kvällstid)
Tel arbete: 018-18 13 08 (dagtid - ring igen om ingen svarar! Jag har undervisning ibland)

N Ä S T A N U M M E R

I förra numret utlovades en recension av WordStar 4.0, men den ser lite svår ut att ordna. För närvarande kan bara få tag i den nya WordStar versionen genom att köpa den direkt från Australien.

Inte heller recensionen av TRANSRECORD fanns med som utlovat. Den kommer istället i nästa nummer som för övrigt kommer ha temat kommunikation och handlar alltså mycket om modem, databaser etc.

Det blir också premiär för en 'Ungdomssida' som kommer innehålla lite enklare tips och program. Vidare så kommer en artikel om hur du bygger om din Microbee till en 6.75 Mhz turbo maskin!

Slutligen så väntar sig förstås redaktionen massor av brev, artiklar och de där programmen som ni har liggande i byrålådan.